Hazard Analysis
RoCam

Team #3, SpaceY
Zifan Si
Jianqing Liu
Mike Chen
Xiaotian Lou

Table 1: Revision History

Date	**Developer(s)**	**Change**
Date1	Shike Chen	Initial draft
Date2	Shike Chen	Detailed draft

Contents

1 Introduction

Hazard is generally defined as a harm or potential harm or effect caused by a system which may lead to personal injury, property damage, environmental damage etc. As an integrated system with both hardware and software components, hazards may arise due to improper design, malfunction of the system, unexpected user behavior, etc. Therefore, it is critical to identify potential hazards and mitigate them in the early stage of the project. In the case of the RoCam project, the potential hazards could cause physical injury of the user or malfunction of a larger system.

2 Scope and Purpose of Hazard Analysis

The hazard analysis will identify any direct or potential hazards caused by the system which includes hazardous materials, design, user safety, etc. The hazard analysis will also address the general hazardous effects of this system which includes hazards effects in the events of malfunction, misuse or environmental damage. However, since the project is open source, any use case that is beyond the scope of this project will not be included in this hazard analysis.

Such hazard analysis will not only used to identify potential hazards caused by the system, but also to minimize risks related in unforeseen cases. The purpose of this hazard analysis is to properly address the potential harm and risks caused by the product to the user and the external environment.

3 System Boundaries and Components

3.1 Hardware Components

3.1.1 Motion control modules

The motion control module consists of a custom-designed PCB board utilizing an STM32 microcontroller. This module is responsible for controlling the movement of the camera gimbal. It receives commands from our Computer Vision module and translates them into precise motor movements in real time.

3.1.2 Camera gimbal

The camera gimbal used in this project is a existing Camera Gimbal designed by the McMaster Rocketry team. Since our team is not responsible for the design of the gimbal, we can only estimate the potential hazards based on similar products in the market.

3.1.3 Jetson Orin Nano

The Jetson Orin Nano is small, lightweight AI computer designed by NVIDIA. It will be used to deploy our computer vision moodule and communicate directly with the motion control module.

3.1.4 camera

The camera is a off the shelf camera that is compatible with the Jetson Orin Nano. It is able to provide 4k video input in real time.

3.2 Software Components

3.2.1 Computer Vision Module

Our computer vision module will be developed in Python. The base architecture will be torch and pandas. It will be accerlerated with TensorRT in Jetson to achieve real time stable performance.

3.2.2 User Interface Module

The user interface module will be developed using React. It will provide a web-based interface for users to interact with the system, view camera feeds, and control the camera gimbal.

4 Critical Assumptions

4.1 Assumptions about the Operating Environment

The system will not operating in any extreme environment (e.g., extreme temperature, acidic environments, etc.). The system will be operating within 10 to 25 degree Celsius. The system will operate with a stable power supply.

4.2 Assumptions about the User

Anyone above 14 years old who uses electronic and exposed to software interface on a daily basis for over 2 years with good intent.

4.3 Assumptions about general of the shelf hardware components

All off-the-shelf hardware components will function as advertised and will be compatible with our system. This includes the camera, gimbal, Jetson Orin Nano etc.

4.4 Assumptions about the software libraries

All software libraries used will not have any unknown security vulnerabilities.

5 Failure Mode and Effect Analysis

Table 2: FMEA

Failure Mode and Effects Analysis **System**: Audio to sheet music generator **Phase/Mode**: System Requirements						
Design Function	**Failure Modes**	**Effects of Failure**	**Causes of Failure**	**Recommended Action**	**SR**	**Ref.**
Generate sheet music	Sheet music flashes on the screen	Potential trigger for users with epilepsy	Incorrect rendering or scrolling through the sheet music	Limit or disable visual flicker; provide user setting	PR-SC1	
	A bad microphone causes notes to be off due to pitch drift	Embarrassment for the user	Microphone exhibits pitch drift (mechanical issue)	Troubleshooting steps; suggest higher-quality microphone	N/A	
Process audio	Audio improperly recorded	Loss of musical performance	Microphone error	Notify user about microphone issue; retry guidance	N/A	
			Signal processing error	Notify user a processing error occurred	FR-SP4	
File functions	Files outside scope of the application are deleted	Possible system failure if critical files removed	Improper file I/O actions	Restrict permissions; sandbox; confirmations	S-P1	
	Files outside scope of the application are modified	System instability / data loss	Improper file I/O actions	Same mitigations as above	S-P1	
Output audio	Overly loud audio emitted	Hearing damage / startle	Miscalibrated volume path	Output limiter; safe defaults; pre-playback check	N/A	
		User is surprised and has a health incident	Miscalibrated volume path	Same as above	N/A	
Application runtime	App crashes unexpectedly	User loses progress	Power outage, OOM error, overheating, malware	Autosave (with consent); crash recovery; logging	PR-RFT3	

6 Safety and Security Requirements

7 Roadmap

Appendix — Reflection

The purpose of reflection questions is to give you a chance to assess your own learning and that of your group as a whole, and to find ways to improve in the future. Reflection is an important part of the learning process. Reflection is also an essential component of a successful software development process.

Reflections are most interesting and useful when they're honest, even if the stories they tell are imperfect. You will be marked based on your depth of thought and analysis, and not based on the content of the reflections themselves. Thus, for full marks we encourage you to answer openly and honestly and to avoid simply writing "what you think the evaluator wants to hear."

Please answer the following questions. Some questions can be answered on the team level, but where appropriate, each team member should write their own response:

1. What went well while writing this deliverable?
2. What pain points did you experience during this deliverable, and how did you resolve them?
3. Which of your listed risks had your team thought of before this deliverable, and which did you think of while doing this deliverable? For the latter ones (ones you thought of while doing the Hazard Analysis), how did they come about?
4. Other than the risk of physical harm (some projects may not have any appreciable risks of this form), list at least 2 other types of risk in software products. Why are they important to consider?